Statistical Properties of Factor Oracles

Jérémie Bourdon, Irena Rusu

Statistical Properties of Factor Oracles

Jérémie Bourdon[1,2] and Irena Rusu[1]

[1] LINA, CNRS UMR 6421 and University of Nantes, France
[2] IRISA INRIA Rennes-Bretagne Atlantique
(Jeremie.Bourdon,Irena.Rusu)@univ-nantes.fr

Abstract. Factor and suffix oracles have been introduced in [1] in order to provide an economic and efficient solution for storing all the factors and suffixes respectively of a given text. Whereas good estimations exist for the size of the factor/suffix oracle in the worst case, no average-case analysis has been done until now. In this paper, we give an estimation of the average size for the factor/suffix oracle of an n-length text when the alphabet size is 2 and under a Bernoulli distribution model with parameter 1/2. To reach this goal, a new oracle is defined, which shares many of the properties of a factor/suffix oracle but is easier to study and provides an upper bound of the average size we are interested in. Our study introduces tools that could be further used in other average-case analysis on factor/suffix oracles, for instance when the alphabet size is arbitrary.
Keywords: indexing structure, average-case analysis, factor recognition, suffix recognition

1 Introduction

Finding a given pattern inside a given text is a classical problem (the *pattern matching* problem) for which many solutions have been proposed until now. A very important class of solutions relies on the use of indexing structures, *i.e.* data structures that allow to store the text, to have a fast access to it and to quickly execute certain operations on data. Suffix arrays, suffix automata, suffix trees are classical structures which can be implemented in linear time with respect to the text size.

Still, these structures require a too important (although linear) amount of space. Several techniques for reducing the memory space needed by index implementation were developed (see [4] for a survey). *Language approximation* is one of these techniques, and factor/suffix oracles (introduced in [1]) are one way to illustrate it. Whereas suffix arrays, suffix automata and suffix trees owe their efficacity to their perfect accuracy when answering to the question "Is the word w a suffix (or a factor) of the stored text?", the factor/suffix oracles are only accurate when they provide the negative answer. The language each of them recognizes is

larger or equal to the set of factors/suffixes (respectively) of the text, but their size is very small. The words accepted by a factor/suffix oracle which are not factors/suffixes of the stored text will be termed *by-products*.

A simple, space economical and linear on-line algorithm to build oracles is given in [1], together with some applications to pattern matching. Other applications to pattern matching, finding maximal repeats and text compression can be found in [8], [9], [10] and [11]. A linear compression algorithm, improving the previous quadratic algorithms proposed in [2] and [3], to transform a suffix tree into an oracle can be found in [16]. Another algorithm, based on Ukkonen's algorithm to build a suffix tree, is given in [5].

Two ideas come easily out from these applications. On the one hand, oracles should be reasonably envisaged when one has to deal with a text mining problem. On the other hand, evaluating precisely the performances of an application that uses oracles is a hard task, especially in the average case. Although theoretical studies have been performed for the maximum number of transitions [1] and the maximum number of by-products [12] for the oracles of an n-length text, no theoretical study exists in the average case (An experimental study was realized in [12] for the number of by-products). As a consequence, no theoretical average-case running-time or memory space analysis exists for any algorithm based on oracles. Moreover, experimentally supported conjectures are still open. This is the case, for instance, for the conjecture claiming that the BOM pattern matching algorithm presented in [1] is optimal in the average.

In this paper, we estimate the average number of transitions (*i.e.* the average space occupancy) of the factor/suffix oracle of an n-length text, when the alphabet size is 2 and under a Bernoulli distribution model with parameter 1/2. In this way, we answer another one of the questions raised in the seminal paper [1] (and raised again in [5]). The first of these questions, concerning the characterization of the language recognized by the factor/suffix oracle, was answered in [13].

The paper is organized as follows. In Section 2 we define the factor/suffix min-oracle (which is the classical factor/suffix oracle) and present its main properties. In Section 3, the factor/suffix short-oracle is introduced and is briefly compared to the factor/suffix min-oracle. In section 4, we investigate local properties of the min- and short-oracles and deduce probabilistic results, that we use in Section 5 to estimate the average space occupancy of a short-oracle, and thus of a min-oracle. Section 6 is the conclusion.

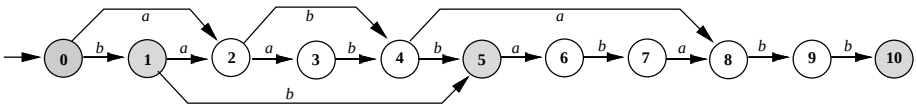

Fig. 1. *The suffix* min*-oracle* $Omin(w)$ *for* $w = baabbababb$*. The final states are grey.*

2 Factor and Suffix (min-)Oracles

Let $w = w_1 w_2 \dots w_n$ be a sequence of length $|w| = n$ on a finite alphabet Σ. Given integers i, j, $1 \leq i, j \leq n$, we denote $w[i \dots j] = w_i w_{i+1} \dots w_j$ and we call this word a *factor* of w (notice that when $j < i$ the resulting factor is by convention the empty word ε). A *suffix* of w is a factor of w one of whose occurrences ends in position n. The i-th suffix of w, denoted $Suff_w(i)$, is the suffix $w[i \dots n]$ and has length $n + 1 - i$. A *prefix* of w is a factor of w one of whose occurrences starts in position 1. The i-th prefix of w, denoted $Pref_w(i)$, is the prefix $w[1 \dots i]$. By convention, the empty word ε is both a suffix and a prefix of w. Say that a suffix of w is *maximal* if it is not identical to w and it is not the prefix of another suffix of w. Say that a suffix of w is *repeated* if it is a factor of $w[1 \dots n-1]$, and *non-repeated* in the contrary case. It is easy to see that a maximal suffix is always a non-repeated suffix, whereas the viceversa is true only for non-repeated *proper* suffixes, *i.e.* distinct from w.

The *factor/suffix oracle* of w is a deterministic automaton which has $n + 1$ states denoted $0, 1, 2, \dots, n$, one *internal transition* $(i, w_{i+1}, i + 1)$ for each state i except n, and at most $n - 1$ *external transitions* denoted (i, w_j, j), for some pairs i, j with $i+1 < j$. Consequently, the factor/suffix oracle of w is *homogeneous*, that is, all the transitions incoming to a given state have the same label. Each state is final in the factor oracle, while only the states ending the spelling of a suffix of w (including the empty one) are final in the suffix oracle (see Figure 1 for the suffix oracle of $w = baabbababb$).

The factor/suffix oracle was introduced in [1] and can be built using an on-line linear algorithm. The algorithm **Build_Oracle** we give here (also proposed in [1]) is quadratic, but more intuitive. In the algorithm, $Omin(w)$ denotes indifferently the factor or suffix oracle.

Figure 1 shows that the factor/suffix oracle can accept words that are not factors/suffixes, *e.g. baabb* which is not a suffix of $w = baabbababb$ but is accepted in the final state 4 of its suffix oracle. These words are called *by-products*.

Algorithm Build_Oracle [1]
Input: Sequence w.
Output: $Omin(w)$.

1. **for** i **from** 0 **to** n **do**
2. create a new state i;
3. **for** i **from** 0 **to** $n-1$ **do**
4. build a new transition from i to $i+1$ by w_{i+1};
5. **for** i **from** 0 **to** $n-1$ **do**
6. let x be a minimum length word whose reading ends in state i;
7. **for** all $\gamma \in \Sigma$, $\gamma \neq w_{i+1}$ **do**
8. **if** $x\gamma$ is a factor of $w' = w[i-|x|+1 \dots n]$ **then**
9. let j be the end position of the first occurrence of $x\gamma$ in w';
10. build a transition from i to j by γ
11. **endif**
12. **endfor**
13. **endfor**

Several important results on oracles have been proved in [1]. Here are the ones which will be needed in the rest of the paper. We denote $poccur(v, w)$ the ending position of the first occurrence of v in w, for each factor v of w.

Lemma 1. [1] *Let w be a word of length n on the alphabet Σ. Then we have:*

(i) *For each state i of $Omin(w)$, there is a unique minimum length word accepted in i, that we note $min_w(i)$.*
(ii) *For each state i of $Omin(w)$, we have $i = poccur(min_w(i), w)$. In addition, $min_w(i)$ is a suffix of every other word accepted in state i.*
(iii) *If $i < j$ are two states of $Omin(w)$ and $\gamma \in \Sigma$, then there exists a transition (i, γ, j) in $Omin(w)$ if and only if we have $j = poccur(min_w(i)\gamma, w)$.*
(iv) *Each factor v of w is recognized by $Omin(w)$ in a state j such that $j \leq poccur(v, w)$.*

For a word u on Σ, let $min(u) = min_u(|u|)$ and notice that if we denote $u = Pref_i(w)$, then $min(u) = min_w(i)$ and all the properties in Lemma 1 may be formulated using $min(u)$ instead of $min_w(i)$.

Remark 1. The algorithm Build_Oracle may be seen as a generic algorithm where the function used to define the word x in step 6 acts as a generator of external transitions. From this perspective, the factor/suffix

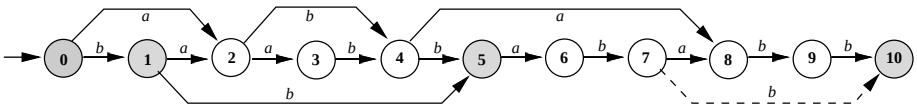

Fig. 2. The suffix short-oracle $Oshort(w)$ for $w = baabbabab$. The final states are grey. The supplementary transition with respect to $Omin(w)$ is dotted.

oracle is the automaton defined by this generic algorithm using the precise function $min()$ as a generator. This is why, in the rest of the paper, the factor/suffix oracle will be called the **factor/suffix min-oracle** (or simply the **min-oracle**) and will be denoted (as we already did) $Omin(w)$.

The best (to the date) estimation of the maximum number of external transitions in a min-oracle was proved in [16].

Lemma 2. [16] *The number of external transitions $ETmin(w)$ of the oracle $Omin(w)$ is upper bounded by the number of maximal suffixes of w.*

3 Factor and suffix short-Oracles

Provided a word u on Σ, denote $short(u)$ the shortest non-repeated suffix of u (by convention, $short(\varepsilon) = \varepsilon$). Then, consider the generic algorithm Build_Oracle in which the generator is now the fonction $short()$. Or, equivalently, step 6 now reads $x = short(Pref_i(w))$, instead of the affectation $x = min(Pref_i(w))$ performed to obtain $Omin(w)$. The resulting homogeneous automaton is denoted $Oshort(w)$ and is called the **short-oracle** of w. Its factor and suffix versions are obtained as for the min-oracle.

Remark 2. For some sequences w, $Omin(w)$ and $Oshort(w)$ are identical, but this is not always the case, since $short(u)$ and $min(u)$ may be different, as is the case for $u = baabbab$: $short(u) = bab$ and $min(u) = bbab$. Then $Oshort(baabbababb)$ has one external transition labeled b leaving state 7 (see Figure 3) because of the occurrence of $babb$ ending in state 10. In opposition, $Omin(baabbababb)$ has no such transition since $bbabb$ has no occurrence ending in a state $j > 7$.

Although possibly different, the min- and short-oracles share many good properties, as shown by the following claim, very close to Lemma 1.

Claim 1. *Let w be a word of length n on the alphabet Σ. Then we have:*

(i) *For each word u, there is a unique shortest non-repeated suffix of u. Consequently $short(u)$ is well-defined.*
(ii) *For each state i of $Oshort(w)$, we have $i = poccur(short(u), w)$ where $u = Pref_i(w)$. In addition, $short(u)$ is a suffix of every other word accepted in state i.*
(iii) *If $i < j$ are two states of $Oshort(w)$ and $\gamma \in \Sigma$, then there exists a transition (i, γ, j) in $Oshort(w)$ if and only if we have $j = poccur(short(u)\gamma, w)$, where $u = Pref_i(w)$.*
(iv) *Each factor v of w is recognized by $Oshort(w)$ in a state j such that $j \leq poccur(v, w)$.*

It is worth noticing here that, although the external transitions of the min- and short-oracles are built according to similar rules and satisfy similar properties (items (iii) in Lemma 1 and Claim 1), it is however much easier to find $short(u)$ than $min(u)$. Indeed, $short(u)$ is simply obtained by considering every suffix of u and testing whether it occurs elsewhere in u, whereas finding $min(u)$ needs to build the min-oracle. As a consequence, it is much easier as well to estimate the number of external transitions in $Oshort(w)$ than in $Omin(w)$. This is why the following result is essential.

Claim 2. *Let w be a sequence and let $ETmin(w)$, $ETshort(w)$ be the number of external transitions in $Omin(w)$ and $Oshort(w)$ respectively. Then we have $ETmin(w) \leq ETshort(w)$.*

4 Probabilities that an external transition exists for binary alphabets

We now focus on random binary sequences issued from an unbiased Bernoulli model $\mathbb{B}$, in which a sequence w on $\Sigma = \{a, b\}$ is produced with probability $p_w = 1/2^{|w|}$. We denote by $\mathbb{B}_n$ the restriction of $\mathbb{B}$ to sequences w of length n.

The two parameters below are of great relevance for our study:

- $pmin_{i \to j}$, where $0 \leq i < j \leq n$, is the probability that there exists a transition from state i to state j in $Omin(w)$.
- $pmin_i$, where $0 \leq i < n$, is the probability that an external transition leaving state i exists in $Omin(w)$. Obviously, the equality $pmin_i = \sum_{j=i+2}^{n} pmin_{i \to j}$ holds.

We first provide exact expressions for the probabilities $pmin_{i\to j}$ and $pmin_i$ when $i = 0$ or $i = 1$. In these simple cases, it is possible to characterize precisely the language of sequences whose min-oracle possesses a transition from state i to state j, when two states i and j are given. An exact formula for the expected probability is then derived. In the general case, such a characterization is no longer possible and we use a method based on Guibas-Odlyzko's equations together with a generating functions methodology to obtain the desired probabilities, as well as their equivalents in the short-oracle.

4.1 Languages viewpoint

Leaving state $i = 0$. First, we study the case of transitions that leave state 0. Let w be a sequence of length n and let j $(1 < j \leq n)$ be an integer. It is obvious that the min-oracle of w possesses a transition from 0 to j if and only if j is the position of first occurrence of a new letter. In the binary case, this means that w is any sequence of one of the languages $a^{j-1}b(a+b)^{n-j}$ or $b^{j-1}a(a+b)^{n-j}$. It is easy to show the following:

Claim 3. *Let j $(1 < j \leq n)$ be an integer. Under the Bernoulli model $\mathbb{B}_n$, we have $pmin_{0\to j} = \frac{1}{2^{j-1}}$ and $pmin_0 = 1 - \frac{1}{2^{n-1}}$.*

Leaving state $i = 1$. Let j $(3 < j \leq n)$ be an integer. Two cases must be considered with respect to the two first letters of the sequence w.

If they are equal, say aa, then there is a transition from state $i = 1$ to state j if, and only if, j is the position of the first occurrence of b in w. The probability of such an event is $1/2^{j-1}$.

If they are distinct, say ab, then there exists a transition from state $i = 1$ to state j if, and only if, the first occurrence of aa ends at position j. This implies that $j > 4$ and w must belong to one of the two languages $\mathcal{L}_a = ab[(b+ab)^\star \cap (a+b)^{j-4}]aa(a+b)^{n-j}$ and $\mathcal{L}_b = ba[(a+ba)^\star \cap (a+b)^{j-4}]bb(a+b)^{n-j}$. In order to deduce the probability for w to belong to $\mathcal{L}_a$ or $\mathcal{L}_b$, we first give the following result.

Claim 4. *The number of sequences of size $J \geq 0$ of the form $(b+ab)^\star$ equals the $(J+1)$-th Fibonacci number F_{J+1} defined recursively by $F_0 = F_1 = 1$, and for all $h > 1$, $F_h = F_{h-1} + F_{h-2}$.*

Previous lemma together with Binet's formula on Fibonacci numbers ($F_J = (\phi^{J+1} - \overline{\phi}^{J+1})/\sqrt{5}$, where $\phi = \frac{1+\sqrt{5}}{2} \approx 1.618$ is the Golden ratio and $\overline{\phi} = \frac{1-\sqrt{5}}{2} \approx -0.618$ its conjugate), allows us to prove the following result.

Claim 5. *Let j, $3 < j \leq n$ be an integer. Under the Bernoulli model $\mathbb{B}_n$, we have*

$$pmin_{1\to j} = \frac{F_{j-3}+1}{2^{j-1}} = \frac{1}{2^{j-1}}\left[1 + \frac{1}{\sqrt{5}}(\phi^{j-2} - \overline{\phi}^{j-2})\right] \tag{1}$$

$$pmin_1 = 1 - \frac{F_{n-1}+1}{2^{n-1}}.$$

We now focus on obtaining asymptotic expressions when i is arbitrary, and need to apply a classical study involving generating functions.

4.2 Generating functions methodology

This section is devoted to a brief presentation of some essential tools from the generating function theory. The reader can refer to [17] for details and supplementary material. After a general approach using an alphabet with an arbitrary number of symbols that is randomly generated by a Bernoulli probabilistic process, we focus on the simpler case of a binary alphabet whose symbols are produced uniformly at random. In this section, Σ is a finite alphabet, $\Sigma^\star$ is the set of all possible words of any length and Σ^+ is the set of all possible words of any length except the empty word ε. For two sequences x and u in $\Sigma^\star$, the function $occ(x,u)$ counts the number of occurrences of motif x in the text u.

Generating functions are very useful tools to study average-case problems on languages. Let $\mathcal{L}$ be a language. The generating function $L(z)$ associated to language $\mathcal{L}$ is defined by $L(z) = \sum_{u\in\mathcal{L}} p_u z^{|u|}$, where p_u is the probability of word u to be produced. In the sequel, we denote by $[z^k]L(z) = \sum_{u\in\mathcal{L}\cap\Sigma^k} p_u$ the coefficient of z^k in $L(z)$, that equals the probability for a word of length k to belong to $\mathcal{L}$.

Consider the following three sets

$$\begin{aligned}
\mathcal{S}_x &= \{u \in \Sigma^\star, occ(x,u) = 0\},\\
\mathcal{T}_x &= \{u \in \Sigma^\star, u = v\cdot x \text{ and } occ(x,u) = 1\},\\
\mathcal{C}_x &= \{u \in \Sigma^\star, \exists v, v' \in \Sigma^+, v\cdot u = v'\cdot v = x\},
\end{aligned}$$

where $v\cdot u$ denotes the concatenation of the two words u and v in this order. These sets are very classical in the so-called Guibas-Odlyzko [6] methodology. The first one, $\mathcal{S}_x$, is the set of words that do not contain x as a factor. The second one, $\mathcal{T}_x$, is the set of words that contain x only as a suffix. Finally, $\mathcal{C}_x$ is the set of suffixes u of x such that x is a suffix of $x\cdot u$. Set $\mathcal{C}_x$ is commonly called the *autocorrelation set* of x.

In the same vein, we define the correlation set $\mathcal{C}_{x,y}$ between two words x and y by $\mathcal{C}_{x,y} = \{u \in \Sigma^\star, \exists v \in \Sigma^\star, v' \in \Sigma^+, x = v \cdot v' \text{ and } y = v' \cdot u\}$.

Sets $\mathcal{S}_x$, $\mathcal{T}_x$ and $\mathcal{C}_x$ are related by the following equalities

$$\mathcal{S}_x \times \Sigma + \varepsilon = \mathcal{S}_x + \mathcal{T}_x \quad \text{and} \quad \mathcal{S}_x \times x = \mathcal{T}_x \times \mathcal{C}_x.$$

By using decomposition properties of memoryless sources, such algebraic decompositions on sets directly translate into equations involving generating functions. By solving the resulting system of equations, one obtains:

Lemma 3 (Guibas-Odlyzko [6]). *The generating functions, denoted respectively $S_x(z)$, $T_x(z)$ and $C_x(z)$, of the sets $\mathcal{S}_x$, $\mathcal{T}_x$, $\mathcal{C}_x$ satisfy*

$$S_x(z) = \frac{C_x(z)/p_x}{D_x(z)} \quad \textit{and} \quad T_x(z) = \frac{z^{|x|}}{D_x(z)},$$

where p_x is the probability of word x to be produced and $D_x(z) = z^{|x|} + (1-z)C_x(z)/p_x$ is a polynom of degree $|x|$.

Thus $S_x(z)$ and $T_x(z)$ are rational functions whose dominant singularities (*i.e.,* the dominant roots of $D_x(z)$) dictate the main order asymptotic term of $[z^k]S_x(z)$ and $[z^k]T_x(z)$. The following lemma may be found in [15].

Lemma 4 (Szpankowski-Regnier [15]). *The coefficients of $[z^k]$ (with $k > 0$) in $S_x(z)$ and $T_x(z)$ satisfy*

$$[z^k]S_x(z) = K_x \rho_x^{-(k+1)} + O(\mu_x^{-k}) \textit{ and } [z^k]T_x(z) = K'_x \rho_x^{-(k-|x|+1)} + O(\mu_x^{-k}),$$

where ρ_x is the root of $D_x(z)$ of smallest modulus, $K_x = \frac{-C_x(1)}{p_x D'_x(\rho_x)}$, $K'_x = \frac{-1}{D'_x(\rho_x)}$ and μ_x is the second modulus of roots of $D_x(z)$.

As an example, it is easy to get the main order term of $pmin_{1\to j}$. In this case, $pmin_{1\to j}$ and the generating function of $\mathcal{T}_{aa}$ are related by $pmin_{1\to j} = \frac{1}{2^{j-1}} + 2[z^j]T_{aa}(z)$. The denominator $D_{aa}(z) = z^2 + 4(1-z)(1+z/2)$ of $T_{aa}(z)$ possesses $\rho_{aa} = 2/\phi$ as dominant root and $\mu_{aa} = |2/\overline{\phi}| \approx 3.236$. Applying Lemma 4 leads to the expected asymptotic expression of $pmin_{1\to j}$ given in equation (1).

In the case of binary Bernoulli unbiased sources, the root ρ_w can be approximated by a quantity depending only on the word length $|x|$.

Claim 6. *Let x be a binary word of length $k > 1$, $s_k = \rho_{a^k}$ and $r_k = \rho_{a^{k-1}b}$. We have:*

(i) $s_k \le \rho_w \le r_k$;
(ii) $r_{k+1} = s_k$;
(iii) *if* $|x| = k > 2$, $\rho_x = 1 + \frac{1}{2^k} + o(1/2^k)$.

4.3 Probabilities of an external transition : general case

Define the two parameters $pshort_{i\to j}$ and $pshort_i$ similarly to $pmin_{i\to j}$ and $pmin_i$, but for $Oshort(w)$.

Now, coming back to the binary case we show how transition probabilities ($pmin_{i\to j}$, $pmin_i$, $pshort_{i\to j}$ and $pshort_i$) can be related to Guibas-Odlyzko languages $\mathcal{S}_x$ and $\mathcal{T}_x$ defined in previous section.

Remark 3. For the sake of simplicity, we deduce in this subsection general expressions *only* for $pmin_{i\to j}$ and $pmin_i$. However, the reader will easily notice that the only property of $Omin(w)$ used in this section is Lemma 1 (iii), and that this property has an equivalent for $Oshort(w)$, namely Claim 1 (iii). Consequently, the reasoning and the results in this part are easily transfered to $Oshort(w)$ (just by replacing $min(u)$ by $short(u)$ appropriately), so as to obtain similar expressions for $pshort_{i\to j}$ and $pshort_i$.

For each letter $m \in \{a, b\}$, notation $\overline{m}$ designates the opposite letter (e.g., $\overline{a} = b$ and $\overline{b} = a$). We now prove the two following claims.

Claim 7. *Let $i < j - 1$. The set $\mathcal{P}_{i\to j,n}$ of all binary words of length n whose oracle possesses an external transition from state i to state j is*

$$\mathcal{P}_{i\to j,n} = \bigcup_{u\in\Sigma^i, m\in\Sigma} u\cdot m\cdot\left(\left(\mathcal{T}_{\min(u)\cdot\overline{m}} \cup \mathcal{C}_{\min(u)\cdot m,\min(u)\cdot\overline{m}}\right) \cap \Sigma^{j-i-1}\right)\cdot\Sigma^{n-j}.$$

In the same vein, it is possible to obtain a similar expression for the transitions leaving a given state.

Claim 8. *The set $\mathcal{P}_{i,n}$ of all binary words of length n whose factor oracle possesses an external transition leaving state i equals*

$$\mathcal{P}_{i,n} = \bigcup_{u\in\Sigma^i, m\in\Sigma} u\cdot m\cdot\Big(\big({}^c\mathcal{S}_{\min(u)\cdot\overline{m}} \cup (\mathcal{S}_{\min(u)\cdot\overline{m}} \cap (\mathcal{C}_{\min(u)\cdot m,\min(u)\cdot\overline{m}}\cdot\Sigma^\star))\big) \cap \Sigma^{n-i-1}\Big),$$

where ${}^cX = \Sigma^\star \setminus X$ denotes the complementary set of X.

Formulas for $pmin_{i\to j}$ and $pmin_i$. It is now obvious to derive expressions for $pmin_{i\to j}$ and $pmin_i$ by means of dominant roots of Guibas-Odlyzko's generating functions. Indeed, $pmin_{i\to j} = \sum_{w\in\mathcal{P}_{i\to j,n}} p_w$ and $pmin_i = \sum_{w\in\mathcal{P}_{i,n}} p_w$. Then, Claims 7 and 8 allow to express these probabilities as particular coefficients of generating functions $T_{\min(u)\cdot\overline{m}}(z)$,

$S_{\min(u)\cdot\overline{m}}(z)$ and $C_{\min(u)\cdot m,\min(u)\cdot\overline{m}}(z)$. The following claim providing asymptotic approximations for the transition probabilities is a direct consequence of Lemma 4.

Claim 9. *Under $\mathbb{B}_n$, the probabilities that an external transition exists satisfy*

$$pmin_{i\to j} = \frac{1}{2^{i+1}} \sum_{u\cdot m\in\Sigma^{i+1}} K'_{\min(u)\cdot\overline{m}} \rho_{\min(u)\cdot\overline{m}}^{-j+i+|\min(u)|+1} + O(1/2^{j-i}),$$

$$pmin_i = \frac{1}{2^{i+1}} \sum_{u\cdot m\in\Sigma^{i+1}} \left(1 - K_{\min(u)\cdot\overline{m}} \rho_{\min(u)\cdot\overline{m}}^{-n+i}\right) + O(1/2^{n-i-1}),$$

where ρ_x, K_x and K'_x are quantities defined in Lemma 4.

Simpler approximations. The two previous expressions are quite ineffective because they involve sums over all possible words of a given length. Now, we show that it is possible to obtain computable approximation formulas for $pmin_{i\to j}$ and $pmin_i$. The approximation involves the probability distribution of the minimum length words, which is defined as follows. Let $M(u) = |\min(u)|$ be the function that associates with any word u the length of its minimum length word. The restriction of $M(u)$ to $\mathbb{B}_i$ is itself a random variable denoted by M_i. Its probability distribution, called in the sequel *probability distribution of minimum length words* is defined by $\text{Prob}\{M_i = k\} = \sum_{u\in\Sigma^i, M(u)=k} \frac{1}{2^i}$.

Claim 10. *Let $\text{Prob}\{M_i = k\}$ be the probability distribution of minimum length words, $\alpha_k = \frac{1}{2^k}$ and $\lambda_k = 1 + \frac{1}{2^k}$. The transition probabilities $pmin_{i\to j}$ and $pmin_i$ satisfy*

$$pmin_{i\to j} = \sum_{k=1}^{i} \text{Prob}\{M_i = k\}\, \alpha_{k+1} \lambda_{k+1}^{-j+i+k+1} + O(1/2^{j-i}),$$

$$pmin_i = 1 - \sum_{k=1}^{i} \text{Prob}\{M_i = k\}\, \lambda_{k+1}^{-n+i} + O(1/2^{n-i-1}).$$

Remark 4. According to Remark 3, $pshort_{i\to j}$ and $pshort_i$ satisfy the same equalities as $pmin_{i\to j}$ and $pmin_i$ in Claim 10, up to $\text{Prob}\{M_i = k\}$ which is replaced by $\text{Prob}\{S_i = k\}$, where $S(u) = |\text{short}(u)|$ is the size of the minimum length non repeated suffix of u and S_i its restriction to $\mathbb{B}_i$.

5 Average space occupancy

The memory requirement for storing the min-oracle of w is the sum of the number of states of $Omin(w)$ (fixed and equal to $n+1$), the number of internal transitions (fixed and equal to n) and the number of external transitions. As an application of our results, we present now an estimation of the average space occupancy (in terms of external transitions) $\mathrm{E}\left[ETmin_n\right]$, where $ETmin(w)$ is the function that counts the number of external transitions of $Omin(w)$ and $ETmin_n$ is its restriction on $\mathbb{B}_n$. This estimation is computable in linear time.

Theorem 1. *Under $\mathbb{B}_n$ (the set of random independently and identically distributed binary words of length n), the average space occupancy $\mathrm{E}\left[ETmin_n\right]$ in terms of external transitions of a* min*-oracle for a word of length n satisfies*

$$\mathrm{E}\left[ETmin_n\right] \leq pmin_0 + pmin_1 + (n-3) - \sum_{k=2}^{n-2} \frac{\gamma_k^{k-1}\lambda_{k+1}^{k-n} - \gamma_k^{n-2}\lambda_{k+1}^{-1}}{1-\gamma_k\lambda_{k+1}} - \sum_{k=2}^{n-2} \frac{\gamma_{k-1}^{k-1}\lambda_{k+1}^{k-n} - \gamma_{k-1}^{n-2}\lambda_{k+1}^{-1}}{1-\gamma_{k-1}\lambda_{k+1}}$$

with $\gamma_k = 1 - \frac{1}{2^k}$ and $\lambda_k = 1 + \frac{1}{2^k}$.

Proof. First notice that the average space occupancy equals the sum of all probabilities of leaving states,

$$\mathrm{E}\left[ETmin_n\right] = \sum_{i=0}^{n-2} pmin_i.$$

It is thus of great interest to obtain a tractable formula for $pmin_i$ and consequently for $\mathrm{Prob}\left\{M_i = k\right\}$, the distribution probability of minimum length words. It is still a challenge to obtain such formulas for min-oracles. Claim 2 proves that the average number $\mathrm{E}\left[ETshort_n\right]$ of external transitions of short-oracles provides an upper bound for the expectation $\mathrm{E}\left[ETmin_n\right]$. Then we concentrate on computing $\mathrm{E}\left[ETshort_n\right] = \sum_{i=0}^{n-2} pshort_i$, where the expression of $pshort_i$ is obtained using Remark 4. We then study the probability distribution of S_i. Then, considering the prefix tree built using all the prefixes of the mirror $w_i \cdots w_1$ of word $w = w_1 \cdots w_i$, $|short(w)|-1$ exactly equals the insertion depth of the i-th prefix in the tree. In [14], Park et al. study the probability distribution

of insertion depth in the case of random words built by an i.i.d. binary source. Applying their results to S_i yields $\text{Prob}\{S_i = k\} = \gamma_k^{n-1} - \gamma_{k+1}^{n-1}$, with $\gamma_k = 1 - 2^{-k}$. Next, we use exact formulas for $pshort_0 = pmin_0$ and $pshort_1 = pmin_1$ and approximations for other probabilities. Finally, it is possible to invert the double sum $\sum_{i=2}^{n-2} \sum_{k=2}^{i}$ into $\sum_{k=2}^{n-2} \sum_{i=2}^{n-2}$ which involves geometric sums leading to the expected result. □

6 Conclusion

In this paper, we provide precise approximations for the probabilities that an external transition exists in the min- and short-oracles. These approximations allow us to study the average space occupancy of these oracles. The main goal of such results is to allow comparing the factor/suffix oracle with other indexing structures such as suffix trees, whose space occupancy closely depends on the number of its internal edges, that is known to be of order $n/\log 2$ (see [7]). Figure 6 compares our bound on the average number of external transitions to the average number of edges of suffix trees. This latter figure suggests a conjecture of $n/3 + 1$ for the average number of external transitions of short-oracles.

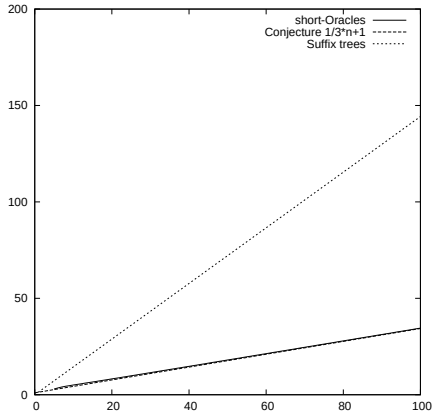

Fig. 3. A comparison of the space occupancy of short-oracles and suffix trees

Notice that one of the main open questions arising when studying oracles concerns the number of words, recognized by an oracle, that are not factor or suffixes. Our results should certainly be helpful since the total number of words recognized by a factor oracle expresses as a sum

$\sum_{k=0}^{n} N_k$, where N_i is the expected number of words recognized in state i. They satisfy $N_0 = 1$ and for all $0 < j \leq n$, $N_j = \sum_{i=0}^{j} pmin_{i \to j} N_i$. It is still a challenge to solve this latter recurrence. Nevertheless, it is quite easy to design a dynamical programming algorithm yielding an upper bound for the expected number of words recognized by a min-oracle, in the same vein of our bound for the expected number of external transitions.

References

1. C. Allauzen, M. Crochemore, M. Raffinot - Factor Oracle: A New Structure for Pattern Matching, In *Proceedings of SOFSEM '99, Theory and Practice of Informatics*, LNCS 1725, 295–310 (1999).
2. G. Assayag, S. Dubnov - Using Factor Oracles for Machine Improvisation, *Soft Computing* 8, 1–7 (2004).
3. L. Cleophas, G. Zwaan, B. W. Watson - Constructing Factor Oracles, In *Proceedings of the Prague Stringology Conference* 2003 (PSC '03), 37–50 (2003).
4. M. Crochemore - Reducing space for index implementation, *Theoretical Computer Science*, 292, 185–197 (2003).
5. M. Crochemore, Lucian Ilie, Emine Seid-Hilmi - The Structure of Factor Oracles, *Int. J. Found. Comput. Sci.* 18(4), 781–797 (2007).
6. L. J. Guibas, A. M. Odlyzko - String Overlaps, Pattern Matching, and Nontransitive Games, *J. Combin. Theory Ser. A*, 30(2), 183–208 (1981).
7. P. Jacquet, W. Szpankowski - Autocorrelation on words and its applications: analysis of suffix trees by string-ruler approach. *J. Combin Theory Ser. A* 66(2), 237–269 (1994).
8. R. Kato - A new full-text search algorithm using factor oracle as index, TR-C185, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan (2003).
9. R. Kato - Finding maximal repeats with factor oracles, TR-C190, Dept. of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan (2004).
10. T. Lecroq, A. Lefebvre - Computing repeated factors with a factor oracle, In *Proceedings of the 11th Australasian Workshop On Combinatorial Algorithms*, L. Brankovic and J. Ryan eds., 145–158 (2000).
11. T. Lecroq, A. Lefebvre - Compror: on-line lossless data compression with a factor oracle, *Information Processing Letters* 83, 1–6 (2002).
12. A. Mancheron - Extraction de motifs communs dans un ensemble de séquences, Ph. D. thesis, University of Nantes, France (2006).
13. A. Mancheron, C. Moan - Combinatorial characterization of the language recognized by factor and suffix oracles, *International Journal of Foundations of Computer Science* 16(6), 1179–1191 (2005).
14. G. Park, H-K Hwang, P. Nicodème, W. Szpankowski - Profile of Tries, in Proceedings of LATIN 2008, LNCS 4957, 1–11 (2008).
15. M. Régnier, W. Szpankowski - On Pattern Frequency Occurrences in a Markovian Sequence. *Algorithmica*, 22(4), 631–649 (1998).
16. I. Rusu - Converting Suffix Trees into Factor/Suffix Oracles, *Journal of Discrete Algorithms* 6(2), 324-340 (2008).
17. Wojciech Szpankowski - Average case analysis of algorithms on sequences, Wiley-Interscience Series in Discrete Mathematics and Optimization (2001).
18. D. Wells - The Penguin Book of Curious and Interesting Mathematics (1997).